



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Sistem

Sistem adalah sekumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik *hardware* maupun *software* yang saling berinteraksi sebagai satu kesatuan untuk mencapai tujuan yang sama (Maniah & Hamidin, 2017:1).

Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan (Anggraeni & Irviani, 2017:1).

Dari pengertian di atas maka dapat disimpulkan bahwa “sistem merupakan sekumpulan komponen yang saling berkaitan untuk mencapai suatu tujuan dengan menerima masukan, dan keluaran pada proses yang dijalankan”.

2.1.2 Pengertian Aplikasi

Aplikasi atau perangkat lunak adalah kumpulan intruksi yang ditujukan kepada komputer. Istilah program dan aplikasi lebih sering disebut untuk menyatakan perangkat lunak. Di kalangan profesional teknologi informasi, istilah program biasa digunakan untuk menyatakan hasil karya mereka yang berupa intruksi-intruksi untuk mengendalikan komputer. Di sisi pemakai, hal seperti itu biasa disebut sebagai aplikasi. (Kadir, 2017:3).

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. (Juansyah, 2015:2).

Dari pengertian di atas maka dapat disimpulkan bahwa “aplikasi merupakan penerapan dari suatu perangkat lunak yang digunakan oleh pemakai atau pengguna untuk melaksanakan suatu fungsinya.



2.2 Teori Judul

2.2.1 Pengertian Metode TOPSIS

Technique For Order Preference By Similarity To Ideal Solution (TOPSIS) merupakan pengambilan keputusan multikriteria yang menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan jarak terpanjang dari solusi ideal negatif untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal (Pratiwi, 2016:159).

TOPSIS adalah salah satu metode pengambilan keputusan multikriteria yang memiliki konsep di mana alternatif yang terpilih merupakan alternatif terbaik yang memiliki jarak terpendek dari solusi ideal positif dan jarak terpanjang dari solusi ideal negatif (Setiawan, 2019:16).

Dari pengertian di atas maka dapat disimpulkan bahwa “metode topsis adalah suatu metode yang mampu menyelesaikan masalah dengan multikriteria, baik pada solusi ideal positif maupun solusi ideal negatif”

Adapun tahapan-tahapan dalam metode TOPSIS yaitu:

1. Membuat matriks keputusan yang ternormalisasi.
2. Membuat matriks keputusan yang ternormalisasi terbobot.
3. Menentukan matriks solusi ideal positif dan matriks solusi ideal negatif.
4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif
5. Menentukan nilai preferensi untuk setiap alternatif

Langkah-langkah untuk menyelesaikan permasalahan menggunakan metode TOPSIS:

1. Membuat matriks keputusan yang ternormalisasi. Metode TOPSIS membutuhkan rating kinerja setiap alternatif C_i pada setiap kriteria K_i yang ternormalisasi.



$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

Dengan:

- i = 1,2,...,m; dan $j=1,2,...,n$.
- r_{ij} = matriks keputusan ternormalisasi.
- X_{ij} = bobot kriteria ke j pada alternatif ke i .
- i = alternatif ke i .
- j = kriteria ke j .

2. Membuat matriks keputusan yang ternormalisasi terbobot

$$\begin{bmatrix} y_{11} & y_{21} & y_{ij} \\ y_{21} & y_{22} & y_{ij} \\ y_{31} & y_{32} & y_{ij} \end{bmatrix}$$

Keterangan :

- W_j adalah bobot kriteria ke- j .
- Y_{ij} adalah elemen dari matriks keputusan yang ternormalisasi.

3. Menentukan matriks solusi ideal positif dan matriks solusi ideal negatif.

$$A^+ = (y_1^+, y_2^+, y_i^+)$$

$$A^- = (y_1^-, y_2^-, y_i^-)$$

Di mana :

- $y_j^+ = \max y_{ij}$, jika j adalah atribut keuntungan.
 $\min y_{ij}$, jika j adalah atribut biaya.
- $y_j^- = \min y_{ij}$, jika j adalah atribut keuntungan.
 $\max y_{ij}$, jika j adalah atribut biaya.

4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif (D^+) dan (D^-) matriks solusi ideal negatif.



$$\sqrt{\sum_{i=1}^m (y_i^2 - y_{ij}^+)^2}$$

$$\sqrt{\sum_{i=1}^m (y_i^2 - y_{ij}^-)^2}$$

Keterangan :

- y^+ adalah elemen dari matriks solusi ideal positif.
- y^- adalah elemen dari matriks solusi ideal negatif.

5. Menentukan nilai preferensi untuk setiap alternatif. Nilai preferensi merupakan kedekatan suatu alternatif terhadap solusi ideal.

$$\frac{D_i^-}{D_i^- + D_i^+}$$

Di mana :

Nilai V_i yang lebih besar menunjukkan prioritas alternatif.

2.2.2 Pengertian Perekrutan

Rekrutmen merupakan proses penarikan, seleksi, untuk mendapatkan karyawan yang efektif dan efisien dalam membantu tercapainya suatu tujuan (Hasibuan, 2016:27).

Rekrutmen adalah tindakan atau proses yang dilakukan oleh perusahaan untuk mendapatkan tambahan karyawan dengan cara melalui tahapan proses seleksi sesuai dengan kebutuhan yang diperlukan perusahaan (Mangkunegara, 2015:33).

Dari pengertian di atas maka dapat disimpulkan bahwa “rekrutmen merupakan proses mencari kandidat atau calon yang memenuhi persyaratan dan juga berkualitas dengan kebutuhan organisasi berdasarkan kriteria yang sudah ditetapkan”.



2.2.3 Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) atau *Decision Support System (DSS)* adalah sebuah sistem yang mampu memberikan kemampuan baik kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur (Marbun & Sinaga, 2018:9).

Sistem Pendukung Keputusan merupakan suatu sistem informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data atau model (Pratiwi, 2016:4).

Dari pengertian di atas maka dapat disimpulkan bahwa “sistem pendukung keputusan merupakan sistem yang berbasis komputer yang setiap prosesnya membantu mengambil keputusan dengan memanfaatkan data yang sudah di bobotkan untuk menyelesaikan suatu keputusan yang tak terstruktur”.

2.2.4 Pengertian Website

Website atau disingkat dengan web merupakan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk digital, baik berupa teks, gambar video, audio dan animasi lainnya yang disediakan melalui jalur koneksi internet (Abdulloh, 2016:1)

Website merupakan kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait di mana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). (Fathurrahman, 2014).



2.2.5 Pengertian Penerapan Metode TOPSIS dalam Pengambilan Keputusan pada Perekrutan Calon Karyawan berbasis Website (Studi Kasus: PT Maju Global Motor Palembang)

Merupakan sebuah aplikasi perekrutan calon karyawan dengan menerapkan metode topsis dalam sistem penilaiannya. Aplikasi ini dapat membantu pihak HRD dan kepala departemen dalam menyeleksi peserta calon karyawan baik dalam tahapan seleksi administrasi, psikotes, bidang keahlian ataupun wawancara. Aplikasi ini juga dapat membantu peserta calon karyawan dalam mengetahui lebih jelas informasi lowongan kerja dan dapat mendaftar dengan melalui sistem tanpa harus datang ke lokasi perusahaan.

2.3 Teori Khusus

2.3.1 Jenis-Jenis Keputusan

Jenis-jenis keputusan dibedakan menjadi tiga macam yaitu keputusan terstruktur, keputusan tidak terstruktur, dan keputusan semi terstruktur (Pratiwi, 2016:5-6).

1. Keputusan terstruktur

Keputusan-keputusan yang berkaitan dengan persoalan yang telah diketahui sebelumnya. Proses pengambilan keputusan seperti ini biasanya didasarkan atas teknik-teknik tertentu dan sudah dibuat standarnya. Kategori keputusan ini juga dapat dikatakan suatu proses jawaban secara otomatis pada kebijakan yang sudah ditentukan sebelumnya. Secara alamiah hampir semua masalah rutin dan berulang memiliki parameter-parameter persoalan yang telah diketahui dan terdefinisi dengan baik, sehingga jawaban atau proses pengambilan keputusan pun bersifat rutin dan terjadwal.

2. Keputusan tak terstruktur

Keputusan-keputusan yang berkaitan dengan berbagai persoalan baru. Keputusan tidak terstruktur biasanya juga berkaitan dengan persoalan yang cukup pelik, karena banyak parameter yang tidak diketahui atau belum diketahui. Oleh karena itu, untuk mengambil keputusan ini biasanya intuisi serta pengalaman seorang pelaku organisasi akan sangat membantu.



Keputusan tak terstruktur, adalah “fuzzy”, permasalahan kompleks di mana tak ada solusi yang mengikutinya. Masalah yang tak terstruktur adalah tak adanya 3 fase proses yang terstruktur. Keputusan tidak terstruktur (*unstructured decision*) bukan merupakan keputusan yang berulang dan rutin. Contohnya adalah memilih sampul depan sebuah majalah, mengontrak manajemen tingkat senior, dan memilih proyek penelitian awal yang akan dilakukan. Tidak ada kerangka atau model yang dapat memecahkan masalah sejenis ini. Bahkan, dibutuhkan banyak sekali pertimbangan dan intuisi. Walaupun demikian, keputusan tidak terstruktur dapat didukung oleh bantuan dari keputusan yang diambil berdasar hasil komputer, yang berfungsi untuk memfasilitasi pengumpulan informasi dari berbagai sumber. Contohnya adalah keputusan untuk pengembangan teknologi baru, pengembangan jenis usaha baru, keputusan untuk bergabung dengan perusahaan lain, perekrutan eksekutif.

3. Keputusan semi terstruktur

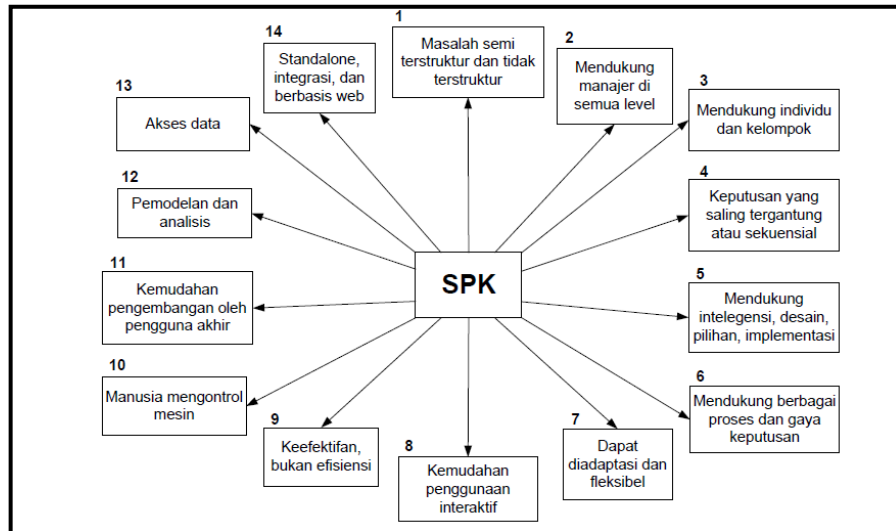
Terdapat beberapa keputusan terstruktur, tetapi tak semua dari fase-fase yang ada. Keputusan semi terstruktur (*semistructured decision*) ditandai dengan dengan peraturan-peraturan yang tidak lengkap untuk mengambil keputusan, dan adanya kebutuhan untuk membuat penilaian serta pertimbangan subjektif sebagai pelengkap analisis data yang formal. Menetapkan anggaran pemasaran untuk suatu produk baru adalah contoh dari keputusan semi terstruktur. Walaupun keputusan seperti ini biasanya tidak dapat secara penuh diotomatisasikan, namun sering didukung dari komputer (*computer-based decision*). Contoh keputusan jenis adalah investasi keuangan, pengevaluasian kredit, penjadwalan produksi, pemberian dana rehabilitasi sekolah, dan pengendalian persediaan.

Sumber: (Pratiwi, 2016:6).



2.3.2 Karakteristik dan Kemampuan Sistem Pendukung Keputusan

Pada Gambar ditunjukkan karakteristik dan kemampuan sebuah sistem pendukung keputusan.



Sumber: (Pratiwi, 2016:7)

Gambar 2.1 Karakteristik dan Kapabilitas SPK

Karakteristik dan kemampuan sebuah sistem pendukung keputusan sebagai berikut:

1. Sistem pendukung keputusan menyediakan dukungan untuk pengambilan keputusan pada keadaan-keadaan semi terstruktur dan tidak terstruktur dengan menggabungkan penilaian manusia dan informasi komputerisasi.
2. Menyediakan dukungan untuk tingkat manajerial mulai dari eksekutif sampai manajer.
3. Menyediakan dukungan untuk kelompok individu, permasalahan yang kurang terstruktur memerlukan keterlibatan beberapa individu dari departemen-departemen yang lain dalam organisasi.
4. Sistem pendukung keputusan menyediakan dukungan kepada independen atau keputusan yang berlanjut.
5. Sistem pendukung keputusan memberikan dukungan kepada semua fase dalam proses pembuatan keputusan *intelligence*, *design*, *choice* dan *implementation*.



6. Sistem pendukung keputusan mendukung banyak proses dan gaya pengambilan keputusan.
7. Sistem pendukung keputusan bersifat adaptif terhadap waktu, sehingga pembuat keputusan harus reaktif dan bisa menghadapi perubahan-perubahan kondisi secara cepat dan merubah sistem pendukung keputusan menjadi *fleksibel* sehingga pengguna dapat menambah, menghapus, mengkombinasikan, merubah dan mengatur kembali terhadap elemen-elemen dasar.
8. Sistem pendukung keputusan mudah digunakan (*user friendly*) dan *fleksibel*.. Kemampuan penggunaan grafik yang tinggi dan bahasa yang mudah dipahami untuk berinteraksi dengan mesin akan menaikkan efektifitas dari sistem pendukung keputusan.
9. Sistem pendukung keputusan menaikkan efektifitas pembuatan keputusan baik dalam hal ketetapan waktu dan kualitas bukan pada biaya pembuatan keputusan atau biaya penggunaan waktu komputer.
10. Pembuat keputusan dapat mengontrol tahapan-tahapan pembuatan keputusan seperti pada tahap *intelligence*, *choice* dan *implementation*, kemudian sistem pendukung keputusan diarahkan untuk mendukung pembuat keputusan bukan menggantikan posisinya.
11. Memungkinkan pengguna akhir dapat membangun sistem sendiri yang sederhana. Sistem yang besar dapat dibangun dengan bantuan dari spesialis sistem informasi.
12. Biasanya model-model digunakan untuk menganalisis situasi pengambilan keputusan. Kapabilitas pemodelan memungkinkan eksperimen dengan berbagai strategi yang berbeda di bawah konfigurasi yang berbeda.
13. Akses disediakan untuk berbagai sumber data, format, dan tipe, mulai dari Sistem Informasi Geografis (SIG) sampai sistem berorientasi-objek.
14. Dapat dilakukan sebagai alat *standalone* yang digunakan oleh seorang pengambil keputusan pada satu lokasi dan di beberapa organisasi sepanjang rantai persediaan. Dapat diintergrasikan dengan DSS lain dan atau aplikasi



lain, dan dapat didistribusikan secara internal dan eksternal dengan menggunakan *networking* dan teknologi *Web*.

2.3.3 Tahapan Pengambilan Keputusan

Alur/proses pemilihan alternatif tindakan/keputusan biasanya terdiri dari langkah-langkah berikut (Pratiwi, 2016:10-11):

1. Tahap *Intelligence*

Suatu tahap proses seseorang dalam rangka pengambil keputusan untuk permasalahan yang dihadapi, terdiri dari aktivitas penelusuran, pendeteksian serta proses pengenalan masalah. Data masukan diperoleh, diuji dalam rangka mengidentifikasi masalah.

2. Tahap *Design*

Tahap proses pengambil keputusan adalah tahap *intelligence* meliputi proses untuk mengerti masalah, mengenali solusi dan menguji kelayakan solusi. Aktivitas yang biasanya dilakukan seperti menemukan, mengembangkan dan menganalisa alternatif tindakan yang dapat dilakukan.

3. Tahap *Choice*

Pada tahap ini dilakukan proses pemilihan di antara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

4. Tahap *Implementation*

Tahap implementasi adalah tahap pelaksanaan dari keputusan yang telah diambil. Pada tahap ini perlu disusun serangkaian tindakan yang terencana, sehingga hasil keputusan dapat dipantau dan disesuaikan apabila diperlukan perbaikan.



2.3.4 Pengertian UML (*Unified Modeling Language*)

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung dan juga salah satu standardisasi bahasa pemodelan yang banyak digunakan didunia industri untuk mendefinisikan pembangunan perangkat lunak yang menggambarkan arsitektur dengan menggunakan teknik pemrograman berorientasi objek (A.S & Shalahudin, 2016:133).

2.3.4.1 Pengertian *Use Case Diagram*

Use Case atau diagram *Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat dan digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi serta siapa saja yang berhak menggunakan fungsi-fungsi itu.

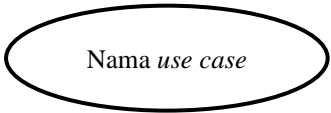
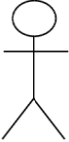


Syarat penamaan pada *use case* adalah nama didefinisikan sesederhana mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.



Berikut adalah simbol-simbol yang ada pada diagram *use case* menurut (A.S & Shalahudin, 2016).

Tabel 2.1 Simbol-simbol *use case diagram*

No.	Simbol	Keterangan
1.	<p><i>Use case</i></p>  <p>Nama <i>use case</i></p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2.	<p>Aktor / <i>actor</i></p>  <p>nama actor</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<p>Asosiasi / <i>association</i></p> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> .
4.	<p>Ekstensi / <i>extend</i></p> <p><<extend>></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; misal arah panah mengarah pada <i>use case</i> yang ditambahkan;

Lanjutan Tabel 2.1 Simbol-simbol *use case diagram*

No	Simbol	Keterangan
5.	Generalisasi / <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
6.	Menggunakan / <i>include</i> / <i>uses</i> <i><<include>></i> <i><<uses>></i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> .

Sumber : (A.S Shalahuddin, 2016:156).

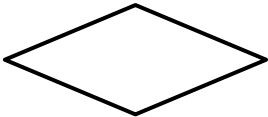


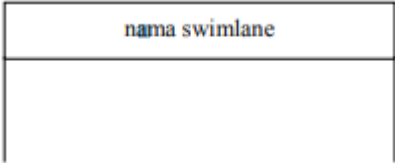
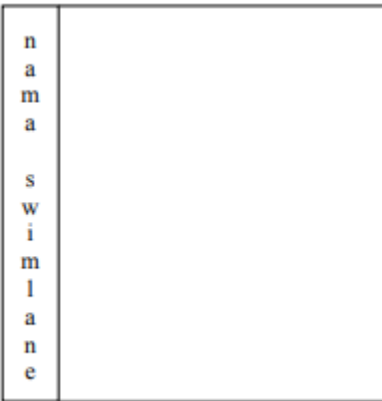
2.3.4.2 Pengertian Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah simbol-simbol yang ada pada diagram aktivitas menurut (A.S & Shalahuddin (2016:161):

Tabel 2.2 Simbol-simbol *activity diagram*

No.	Simbol	Keterangan
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja

Lanjutan Tabel 2.2 Simbol-simbol *activity diagram*

No	Simbol	Keterangan
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6.	<i>Swimlane</i>  atau 	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : (A.S Shalahuddin, 2016:162).



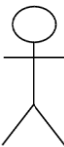
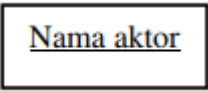

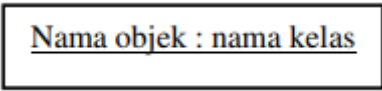
2.3.4.3 Sequence Diagram

Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek”.


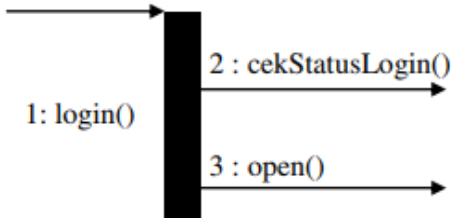


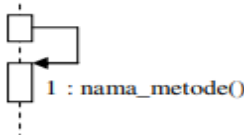
Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram *sequence* menurut (A.S Shalahuddin, 2016:165).


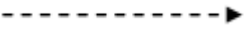
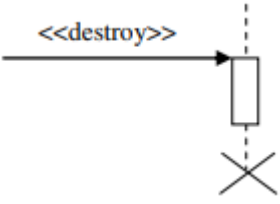
Tabel 2.3 Simbol-simbol *sequence diagram*

No.	Simbol	Keterangan
1.	<p>Aktor</p>  <p>nama aktor atau</p>  <p>tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.
2.	<p>Garis Hidup / lifeline</p> 	Menyatakan kehidupan suatu objek
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan

Lanjutan **Tabel 2.3** Simbol-simbol *sequence diagram*

No	Simbol	Keterangan
4.	Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  Maka <i>cekStatusLogin()</i> dan <i>open()</i> dilakukan didalam metode <i>login()</i> . Aktor tidak memiliki waktu aktif
5.	Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi

Lanjutan **Tabel 2.3** Simbol-simbol *sequence diagram*

No	Simbol	Keterangan
7	Pesan tipe <i>send</i> 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8	Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber : (A.S & Shalahuddin, 2016:165).

2.3.4.4 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

a. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.



b. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

c. Kelas yang diambil dari pendefinisian use case (*controller*)

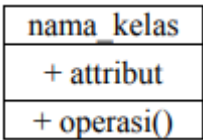
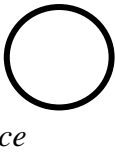


Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

d. Kelas yang diambil dari pendefinisian data (*model*)


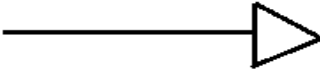

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada diagram kelas menurut (A.S & Shalahuddin, 2016:141):

Tabel 2.4 Simbol-simbol *class diagram*

No.	Simbol	Keterangan
1.		Kelas pada struktur sistem
2.		Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.		Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.		Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Lanjutan **Tabel 2.4** Simbol-simbol *class diagram*

No	Simbol	Keterangan
5.	Agregasi / aggregation 	Relasi antarkelas dengan makna semua- bagian (<i>whole-part</i>)
6.	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
7.	Kebergatungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas

Sumber : (A.S & Shalahuddin, 201;146).

2.3.5 Pengertian *Extreme Programming*

Extreme programming merupakan salah satu metodologi dalam rekayasa perangkat lunak dan juga merupakan satu dari beberapa *agile software development methodologies* yang berfokus pada *coding* sebagai aktivitas utama di semua tahap pada siklus pengembangan perangkat lunak (Kosasi, 2015).

Extreme Programming (XP) adalah metode pengembangan perangkat lunak yang sederhana dan mencakup salah satu metode tangkas yang dipelopori oleh Kent Beck, Ron Jeffries, dan Ward Cunningham. XP adalah salah satu metode tangkas yang paling banyak digunakan dan menjadi pendekatan yang sangat terkenal. Tujuan XP adalah tim yang terbentuk antara kursus berukuran kecil hingga menengah, tidak perlu menggunakan tim besar. Hal ini dimaksudkan untuk mengatasi persyaratan yang tidak jelas dan perubahan persyaratan dengan sangat cepat (Ependi et al., 2014).

Dari pengertian di atas maka dapat disimpulkan bahwa *Extreme programming* (XP) merupakan salah satu model dari *agile software* yang digunakan untuk kebutuhan pengembangan pada sistem yang akan dibuat.



2.4 Teori Program

2.4.1 Pengertian HTML

Hypertext Markup Language (HTML) adalah sebuah bahasa untuk menampilkan konten di web. HTML sendiri adalah dokumen teks yang dapat di edit editor teks apapun. Dokumen HTML punya beberapa elemen yang dikelilingi oleh tag-teks yang dimulai dengan tanda < dan dengan tanda >. (Edy Winarno ST, Ali, & Commu, 2014).

Hypertext Markup Language (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman *web* dan menampilkan berbagai informasi didalam sebuah *browser* internet. (Setiawan, 2015).

2.4.2 Pengertian Xampp

XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket XAMPP sudah terdapat Apache (*web server*), MySQL (*database*), PHP (*server side scripting*), Perl, FTP Server, *PhpMyAdmin* dan berbagai pustaka bantu lainnya (Wahana, 2014:72)

XAMPP adalah perangkat lunak bebas (*free software*) yang mendukung banyak system operasi. Merupakan kompilasi dari berbagai program (Haqi & Heri, 2019).

2.4.3 Pengertian Database

Menurut Abdulloh (2018:103), DBMS yaitu sistem perangkat lunak yang menyediakan layanan bagi user untuk membuat, mengontrol dan mengakses *database*.(Abdulloh, 2018:103).

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. (A.S & Shalahudin, 2016:44).



2.4.4 Pengertian PHP

PHP Hypertext Preprocessor, adalah sebuah bahasa pemrograman web berbasis *server* yang mampu memarsing kode PHP dari kode web dengan ekstensi *.php*, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client* (*browser*). (Edy Winarno ST, Ali, & Commu, 2014).

PHP merupakan bahasan pemrograman *script server-side* yang didesain untuk pengembangan web. Disebut bahasa pemrograman server side dikarenakan *php* dapat diproses pada Komputer server. (Madcoms, 2016)

2.4.5 Pengertian CSS

Cascading Style Sheets (CSS) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahas markup. (Setiawan, 2015).

CSS (*Cascading Style Sheet*) biasa digunakan pada dokumen *web* dan digunakan untuk mengatur tampilan elemen-elemen HTML pada layar, kertas dan bahkan media lain. (Kadir, 2018:143).

2.4.6 Pengertian MYSQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). (Hendry, 2015)

MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Kelebihan dari *MySQL* adalah gratis, selalu di-*update* dan banyak form yang memfasilitasi para pengguna jika memiliki kendala. *MySQL* juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah (Hidayatullah & Kawistara, 2014: 180).



2.4.7 Pengertian *JavaScript*

JavaScript adalah bahasa scripting client side yang sangat populer. Hampir semua programmer *web* menggunakan *JavaScript* untuk memberi efek pemrograman di halaman. (Edy Winarno ST, Ali, & Commu, 2014).

JavaScript adalah salah satu platform yang menggunakan teknik *client side* yang cukup dikenal dan mampu berjalan di banyak *platform* sistem operasi, termasuk sistem operasi *mobile*. (Setiawan, 2015).

2.4.8 Pengertian *Bootstrap*

Bootstrap merupakan salah satu *framework* CSS yang paling populer dari sekian banyak *framework* CSS yang ada. *Bootstrap* memungkinkan desain sebuah web menjadi reponsif sehingga dapat dilihat dari berbagai macam ukuran device dengan tampilan tetap menarik. (Kaban, 2019:261).

Bootstrap merupakan *framework* untuk membangun desain web secara resposif, artinya tampilan web yang dibuat oleh *bootstrap* akan menyesuaikan ukuran *layer* dan *browser* yang kita gunakan baik di desktop, tablet maupun *mobile device*. Dengan *bootstrap* kita juga bisa membangun web dinamis ataupun statis. (Alatas, 2013).

2.4.9 Pengertian *Blackbox*

A.S Rosa dan M. Shalahuddin (2015:275) menjelaskan, *Black Box Testing* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. (A.S Rosa dan M. Shalahuddin, 2015)



2.5 Referensi Jurnal

Penelitian yang dilakukan oleh Ependi *et al.*, (2014) metode pengembangan perangkat lunak yaitu *Extreme Programming* (XP) bertujuan untuk mengurangi biaya perubahan perangkat lunak, dalam metodologi pengembangan sistem tradisional, persyaratan sistem yang ditentukan dalam tahap pertama dari pengembangan proyek dan tidak berubah, yang berarti bahwa biaya untuk kebutuhan perubahan yang terjadi pada tahap selanjutnya akan menjadi sangat mahal. Maka hasil dari penelitian ini untuk mengetahui bagaimana kelemahan dan kekuatan dari metode XP membuat sebuah istilah akuntansi aplikasi pada *smarthphone*.

Khairul Imtiha *et al.*, (2017) dalam penelitian di SMK Bangun Bangsa menemukan sistem penggajian yang masih belum terkomputerisasi yang menimbulkan kesalahan pada proses penghitungan gaji bersih para guru, Perancangan sistem informasi ini menggunakan konsep *Agile Software Development* dengan metodologi *Extreme Programming* (XP), diharapkan dengan adanya sistem informasi penggajian pada SMK Bangun Bangsa yang terkomputerisasi akan mempermudah Bendahara dalam proses perhitungan kehadiran guru, dan rekapitulasi laporan perhitungan gaji.

Yessica Siagian (2018) melakukan penelitian bahwa ada hal yang paling penting bagi perusahaan untuk memperoleh calon karyawan baru yang kompeten. Untuk memecahkan permasalahan tersebut diperlukan teknik analisis yang dapat membantu manajemen perusahaan dalam pengambilan keputusan untuk menentukan karyawan baru di PT Indah Jaya. Metode yang digunakan pada penelitian ini adalah metode TOPSIS, melalui penerapan metode ini peneliti dapat membandingkan setiap calon karyawan berdasarkan kriteria yang menentukan calon karyawan dianggap memenuhi kualifikasi.

Adil Setiawan (2019) dalam melakukan penelitian bahwa ada SMP Muhammadiyah 57 Medan masih sangat dibutuhkan tenaga pengajar yaitu tenaga guru matematika yang profesional. Oleh karena itu perlu dibuat sebuah aplikasi yang mampu menentukan seleksi penerimaan calon tenaga guru matematika profesional yang optimal dapat membantu manajemen sekolah dalam mengambil



keputusan yang lebih objektif. Dengan menggunakan metode TOPSIS yang menggunakan multikriteria atau kriteria yang banyak dalam mengambil keputusan dapat memudahkan manajemen sekolah dalam menyeleksi berkas peserta yang layak dijadikan guru matematika yang profesional.

Hertyana (2019) menyebutkan bahwa dalam menggunakan metode TOPSIS, output dari hasil penelitian memenuhi tujuan karena hasil perhitungan dapat lebih akurat dan lebih cepat dalam menghasilkan informasi yang dapat digunakan sebagai pendukung keputusan dalam suatu perusahaan, data karyawan yang kompeten, proses yang panjang diperlukan karena tidak mudah bagi perusahaan untuk memilih calon karyawan berdasarkan kriteria yang diinginkan oleh perusahaan. Tujuan dari penelitian ini adalah untuk membantu dan mengembangkan pilihan karyawan baru untuk suatu perusahaan.